Adversarial Learning for Discourse Rhetorical Structure Parsing

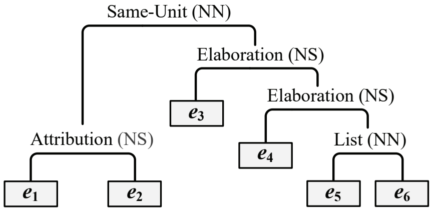
Longyin Zhang[1,2], **Fang Kong**[1,2]*, **Guodong Zhou**[1,2]
1. Institute of Artificial Intelligence, Soochow University, China
2. School of Computer Science and Technology, Soochow University, China
lyzhang9@stu.suda.edu.cn
{kongfang,gdzhou}@suda.edu.cn

Abstract

Text-level discourse rhetorical structure (DRS) parsing is known to be challenging due to the notorious lack of training data. Although recent top-down DRS parsers can better leverage global document context and have achieved certain success, the performance is still far from perfect. To our knowledge, all previous DRS parsers make local decisions for either bottom-up node composition or top-down split point ranking at each time step, and largely ignore DRS parsing from the global view point. Obviously, it is not sufficient to build an entire DRS tree only through these local decisions. In this work, we present our insight on evaluating the pros and cons of the entire DRS tree for global optimization. Specifically, based on recent well-performing top-down frameworks, we introduce a novel method to transform both gold standard and predicted constituency trees into tree diagrams with two color channels. After that, we learn an adversarial bot between gold and fake tree diagrams to estimate the generated DRS trees from a global perspective. We perform experiments on both RST-DT and CDTB corpora and use the original Parseval for performance evaluation. The experimental results show that our parser can substantially improve the performance when compared with previous state-of-the-art parsers.

1 Introduction

As the main linguistic theory on discourse rhetorical structure (DRS), Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) describes an article as a discourse tree (DT). As illustrated in Figure 1, each leaf node of the tree corresponds to an Elementary Discourse Unit (EDU), and relevant leaf nodes are connected by relation and nuclearity (`nucleus` (N) or `satellite` (S)) tags to form high-layer discourse units (DUs), where the `nucleus` is considered more important than the `satellite`. Since the RST structure can well describe the organization of an article, it has been playing a central role in various down-stream tasks like summarization (Xu et al., 2020), text categorization (Ji and Smith, 2017), and so on.



[e_1: In fact,] [e_2: Budget indicated] [e_3: it saw some benefit] [e_4: to staying involved in these programs,] [e_5: in which renters earn frequent-flier miles] [e_6: and fliers can get car-rental discounts.] wsj_2394

Figure 1: An example RST-style discourse tree.

With the release of various discourse corpora, text-level DSR parsing has been drawing more and more attention in the last decade. However, since the corpus annotation is usually time-consuming, existing DRS corpora are much limited in size. For example, the English RST-DT (Carlson et al., 2001) corpus only contains 385 WSJ articles, and the Chinese CDTB (Li et al., 2014b) corpus only contains 500 newswire articles. In this situation, previous studies usually rely on multifarious hand-engineered features (Hernault et al., 2010; Feng and Hirst, 2014; Ji and Eisenstein, 2014; Li et al., 2014a, 2016; Braud et al., 2017). And all these systems perform DRS parsing in a bottom-up fashion. Until recently, some researchers turn to top-down DRS parsing (Lin et al., 2019; Zhang et al., 2020; Kobayashi et al., 2020) to explore the potential capabilities of data-driven models. Nevertheless, text-level DRS parsing is still challenging and worthy of in-depth exploration.

Theoretically, in supervised learning, annotated

*Corresponding author

*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing*, pages 3946–3957
August 1–6, 2021. ©2021 Association for Computational Linguistics

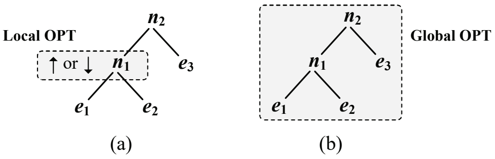


Figure 2: Local and global optimization of DRS trees.

data corpora can provide neural models with specific learning objectives, and the corpus size limitation will weaken the learning of these goals. To mitigate this problem, we researchers need (i) an efficient model to better learn from the limited data and (ii) more high-quality training objectives to enhance the model learning. Existing studies on text-level DRS parsing show that

- Compared with bottom-up DRS parsers, recent top-down frameworks can better leverage global document context and have achieved promising results in text-level DRS parsing (Zhang et al., 2020; Kobayashi et al., 2020).

- All previous studies produce their DRS parsers with local decisions made at each time step for either bottom-up node composition or top-down split point selection (Figure 2 (a)), and no global decisions are made for the entire DRS structure (Figure 2 (b)). Therefore, it is difficult for them to achieve global optimization. Although some studies (Braud et al., 2017; Mabona et al., 2019) leverage "beam-search" to traverse the solution space to find the optimal parsing route, the algorithms are time-consuming to some extent.

Considering the above-mentioned status quo, in this work, we study a global optimization method based on the well-performing top-down parsers. For model structure, we take the top-down parser of Zhang et al. (2020) as our baseline system and make some improvements to it. For global optimization, we first utilize a novel strategy to transform both gold standard and predicted DRS trees into tree diagrams with two color channels. After that, an LSGAN-based adversarial bot is structured between gold and fake tree diagrams as an examiner for global estimation and optimization. Experimental results on the RST-DT and CDTB corpora show that our approaches are effective.

2 Related Work

In the literature, previous studies on RST-style DRS parsing mainly consist of two categories, i.e., bottom-up and top-down frameworks.

For the first category, early studies on DRS parsing heavily relied on hand-crafted features and linguistic characteristics (Hernault et al., 2010; Joty et al., 2013; Feng and Hirst, 2014). During the past decade, more and more researchers turned to data-driven approaches, and some effective strategies were proposed to adapt to the small-scale data corpora. Among these studies, (Ji and Eisenstein, 2014; Li et al., 2014a, 2016; Mabona et al., 2019) used some trivial features as auxiliaries in their data-driven systems; Braud et al. (2016; 2017) harnessed task supervision from related tasks, alternative views on discourse structures, and cross-lingual data to alleviate the data insufficiency problem; Wang et al. (2017) introduced a two-stage parser to first parse a naked tree structure and then determine rhetorical relations for different discourse levels to mitigate data sparsity; Yu et al. (2018) employed both syntax information and discourse boundaries in their transition-based system and achieved good performance.

For the second category, some researchers (Lin et al., 2019; Liu et al., 2019; Zhang et al., 2020; Kobayashi et al., 2020) turned to top-down frameworks to tap the potential capabilities of data-driven models. Among them, (Lin et al., 2019; Liu et al., 2019) have achieved certain success in sentence-level DRS parsing. Nevertheless, due to the long-distance dependency over the discourse, text-level DRS parsing remains challenging. To alleviate this problem, Zhang et al. (2020) proposed a top-down architecture tailored for text-level DRS parsing. Kobayashi et al. (2020) used contextualized word representation and proposed to parse a document in three granularity levels for good performance.

In the past decade, GANs have achieved great progress in NLP (Wu et al., 2019; Elazar and Goldberg, 2018; Chen and Chen, 2019; Zou et al., 2020). However, to our knowledge, there is still no research on adversarial learning in DRS parsing so far. In this work, we explore to adversarially train a discriminator to estimate the quality of the entire DRS tree for global optimization. Notably, we propose to transform each DRS tree into a continuous tree diagram, and thus our adversarial method does not suffer from the "discrete data" problem.

3 Baseline Top-Down Architecture

In this section, we give a brief introduction to our baseline system, the top-down parser of Zhang et al. (2020), and make some improvements to it. The parsing process is illustrated in Figure 3.

Hierarchical Split Point Encoding. For split point representation[1], Zhang et al. (2020) introduced a hierarchical RNN-CNN architecture in their paper. Firstly, they use an attention-based GRU encoder to encode each EDU, obtaining e_i. Then, the obtained EDU vectors are fed into another BiGRU for context modeling, as shown in Figure 3. Next, a CNN net with a window size of 2 and a stride size of 1 is built for each window of EDUs in the discourse for split point encoding. To our knowledge, Zhang et al. (2020) produced dummy split points at both ends of a discourse. Since the dummy split points do not participate in the split point selection process, they could be redundant. Here, we try to simplify the parsing procedure with the dummy split points discarded, as shown in Figure 3. Following previous work (Yu et al., 2018; Kobayashi et al., 2020), we also splice the sentence- and paragraph-level boundary feature vectors to the representation of split points to enhance the encoder model.

Top-Down Split Point Ranking. After achieving split point representations, an encoder-decoder is used to rank the split points, as shown in Figure 3. During encoding, the previously obtained split point vectors are taken as input to the BiGRU encoder, obtaining $H_0, \ldots, H_{n-2}$. During decoding, a uni-directional GRU with an internal stack is used to control the split point ranking process. Initially, the stack contains only one element, i.e., indexes of the boundary split points in the discourse. Notably, since we do not add dummy split points in this parser, we allow patterns like (τ, τ) to appear in the stack. At the j-th step, the tuple (B, E) is popped from the stack and we enter the concatenated $c_j = (H_\mathrm{B}; H_\mathrm{E})$ into the decoder for d_j.

After that, a biaffine function (Dozat and Manning, 2017) is built between the encoder and decoder outputs for split point ranking. Different from (Zhang et al., 2020), all split points in the interval $[\mathrm{B}, \mathrm{E}]$ are selectable in this work. At the step j, we calculate the attention score between H_i and d_j as:

$$s_{j,i} = H_i^{\mathrm{T}} W d_j + U H_i + V d_j + b \tag{1}$$

where W, U, V, b are model parameters and $s_{j,i} \in \mathbb{R}^k$ denotes the score of the i-th split point over different categories (for split point ranking, k equals 1). With this attention function used, at each time step, split position with the highest score is selected as the split point and the original text span is split into two adjacent text spans. Meanwhile, newly generated text spans with unselected split points are pushed onto the stack for following steps, as shown in Figure 3. In this way, a DRS tree is built after 5 iterations with the split points $(1, 0, 2, 3, 4)$ detected in turn.

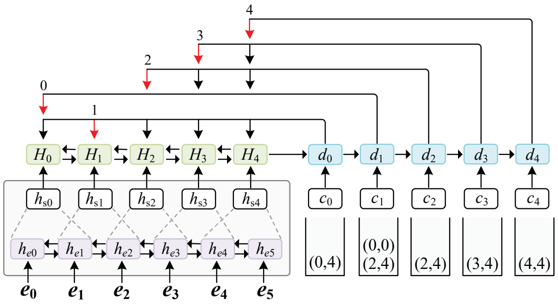


Figure 3: Neural architecture of the encoder-decoder.

To our knowledge, Zhang et al. (2020) use three biaffine classifiers in their parser for structure, nuclearity and relation prediction, respectively. Considering the differences between the three learning objectives, using three independent classifiers could weaken the "Full" performance. To alleviate this problem, we combine nuclearity and relation tags into N-R tags and only use two classifiers for DRS parsing. Therefore, for N-R prediction, the category number k equals 41 and 46 for the RST-DT and CDTB corpus respectively.

4 Adversarial Learning for DRS Parsing

This section introduces the proposed adversarial learning method which consists of two parts: graphical representation of gold and fake DRS trees and the adversarial model learning process.

4.1 Graphical Representation of DRS Trees

In this study, we aim to learn from the entire DRS tree to optimize our model from a global perspective. Usually, our computer understands DRS trees in two ways: either language description or graphical representation. Since tree diagrams can reflect the structural features more intuitively and are easy for machines to understand, we explore graphical representation of DRS trees in this work.

For gold standard trees, we propose to transform each tree into multi-pattern matrices which

[1] The split position between any two neighboring EDUs is called the split point.

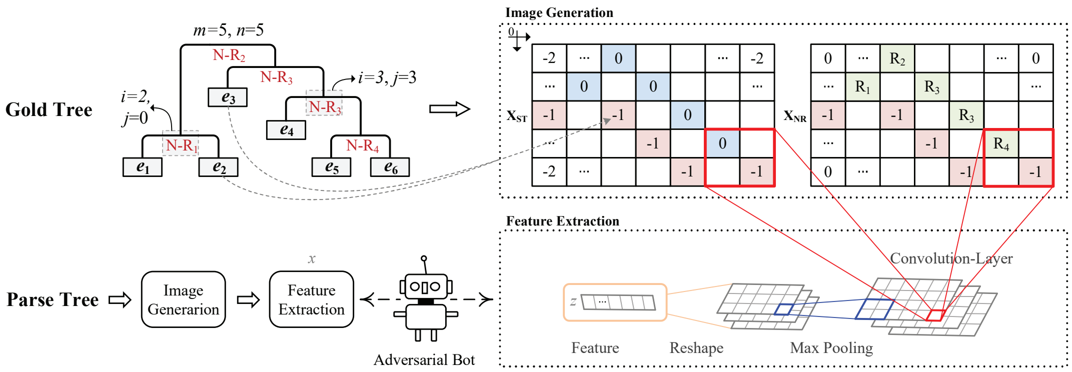


Figure 4: Graphical representation of DRS structure for adversarial learning of text-level DRS parsing.

is similar to a low resolution image with two color channels (i.e., the structure (ST) and nuclearity-relation (NR) channels). Formally, given a DRS tree of height m with n split points, each split point corresponds to a specific non-leaf node in the tree, and we construct two matrices, X_{ST} and X_{NR}, of size $m \times (n+2)$ corresponding to the two color channels, as shown in Figure 4. (i) For the ST channel, all the elements in the matrix X_{ST} are initialized[2] to -2. With the upper left corner of the matrix as the origin of the coordinate axis, given the split point j at the i-th tree layer (top-down direction), we directly set the element at (i-1, j+1) by zero. Besides, if the left span of the split point is an EDU, then we set the element at (i, j) by -1, and the right span is processed in a similar way. With this method, we can recursively construct the tree diagram from top to down. Additionally, some EDU positions are actually shared in the matrix, and this does not affect the understanding of these nodes. For the example in Figure 4, although e_2 and e_3 share a same position in the ST channel, the following two patterns in the matrix can still reveal an accurate representation of each node:

$$\mathcal{N}_1 : \begin{bmatrix} 0 & -2 \\ -2 & -1 \end{bmatrix} \quad \mathcal{N}_2 : \begin{bmatrix} -2 & 0 \\ -1 & -2 \end{bmatrix} \tag{2}$$

(ii) For the NR channel, we set the positions representing non-leaf nodes to specific N-R labels and the positions of leaf nodes to -1 and other non-node positions to zero.

For the automatically parsed trees, we directly use our model outputs to build the tree diagram with two color channels, X'_{ST} and X'_{NR}. And the two matrices of size $m \times (n+2)$ are initialized with zero. (i) For the ST channel, as stated before, a set of attention weights are assigned to the encoder outputs during pointing and a split point is selected according to the weights. Obviously, each split point corresponds to a group of attention weights (after log-softmax). Therefore, we directly add these n-dimensional attention weights of each split point in the i-th tree layer (top-down direction) to the i-th line of X'_{ST}. Notably, the first and last columns of the matrices are actually placeholders initialized with unlearnable scalars representing leaves or non-node positions, so we only add the split point attention weights to the range from 1 to n in each row. (ii) For the NR channel, we simply replace these elements corresponding to split points in X'_{ST} with predicted N-R labels[3] and other elements keep the same as X_{NR}. Alternatively, only the replaced elements in the matrix X'_{NR} are learnable, while other positions serve as static features in the image. In this way, the model outputs are also abstracted as a tree diagram with two color channels.

Through the above methods, we achieve graphical representation for both gold standard and automatically predicted DRS trees. And the graphical representation can provide our model with a global perspective, which makes the global optimization (Subsection 4.2) of DRS parsing possible.

4.2 Adversarial Model Learning

For model learning, we have two goals: (i) learning of DRS parsing at each time step for local optimization and (ii) learning an adversarial bot to evaluate

[2] We set these non-node positions to -2 in two reasons: (i) we apply a log-softmax function to the attention weights for split point ranking with the output ranging $(-\infty, 0]$; (ii) we simply set the non-node positions by -2 to distinguish them from the leaf nodes marked with -1.

[3] Here, we need to map the attention score, $s_{j,i} \in \mathbb{R}^k$, to a specific N-R label. Since the argmax function does not support gradient calculation, we give an alternative solution: $L_{j,i} = \text{F}_{sigmoid}(w_l \cdot s_{j,i} + b_l) \times K$, where K is the number of N-R labels and $L_{j,i} \in \mathbb{R}^1$ is the learnable N-R label.

the pros and cons of the entire tree for global optimization. For the first goal, we use two negative log-likelihood loss terms to optimize the parsing model. For split point ranking, we use L_s to maximize the probability of correct split point selection at each decoding step. For N-R prediction, given the selected split point, we use L_{nr} to maximize the probability of correct N-R labeling for the split point. Since the convergence speeds of the two loss terms are different, we add two loss weights before the loss terms to balance the model training as:

$$L_{\text{DRS}} = \alpha_1 L_s + \alpha_2 L_{nr} \tag{3}$$

For the second goal, we explore to learn from the entire DRS tree for global optimization. To that end, we produce an adversarial bot in our parser to estimate the generated DRS tree diagrams, as shown in Figure 4. Since the composition and sources of gold and generated tree diagrams are completely different, we use two isomorphic feature extractors to understand the two kinds of images separately. For feature extraction, based on such a 2D image-like representation, we perform convolution on every $3 \times (n+2)$ window to dig out the structural details of the entire tree:

$$\varrho_{win}^{(f)} = \mathrm{F}_{relu}(\mathbf{w}^{(f)} \cdot X_{win} + b^{(f)}) \tag{4}$$

Then we perform max-pooling in each nonoverlapping 3×1 window for feature extraction, and the resulting matrices are reshaped as $\varrho \in \mathbb{R}^{1 \times D}$ to serve as the distributed representation of the tree.

In this work, we do not just need an excellent discriminator expert in classification, we need the adversarial nets to continuously give feedback to our parsing model even when the generated trees are correctly classified. On this basis, we leverage Least Squares Generative Adversarial Network (LSGAN) (Mao et al., 2017) as our adversarial bot which has proven to perform more stable and face less problem of vanishing gradients than the original GAN. Formally, our adversarial nets consist of two parts: (i) a generative net G to capture the data distribution p_z over the training data X and (ii) a discriminative net D to estimate the probability that a sample comes from X rather than p_z. On this basis, given the distributed representation of the gold tree x and fake tree z, we formulate the loss functions as follows:

$$\begin{aligned}\min_D V(D) = &\frac{1}{2}\mathbb{E}_{x \sim p_{data}(x)}[(D(x) - b)^2] \\ &+ \frac{1}{2}\mathbb{E}_{z \sim p_z(z)}[(D(G(z)) - a)^2]\end{aligned} \tag{5}$$

$$\min_G V(G) = \frac{1}{2}\mathbb{E}_{z \sim p_z(z)}[(D(G(z)) - c)^2] \tag{6}$$

Similar to Mao et al. (2017), we set $a = 0$ and $b = c = 1$ to make G generate samples as real as possible. Technically, the generator G consists of the parsing model and the feature extractor for fake trees, and the discriminator is an MLP (In: feature size (ϵ), Hidden: $\epsilon/2$, Out: 1) without the sigmoid activation function. Therefore, when learning G, parameters of the parsing model and the feature extractor for fake trees are updated. Likewise, parameters of the discriminator and the feature extractor for real trees are learned when tuning D.

At this time, we have a traditional loss term to train the top-down parser at each splitting step and two adversarial loss terms to estimate the entire DRS tree for global optimization. It is worth mentioning that we first optimize the L_{DRS} for 7 epochs to warm up the model parameters, and then the adversarial nets join the training process for global optimization of DRS parsing.

5 Experimentation

5.1 Experimental Settings

Datasets. Following our previous work (Zhang et al., 2020), we utilize both the English RST Discourse Treebank (RST-DT) (Carlson et al., 2001) and the Chinese Connective-driven Discourse TreeBank (CDTB) (Li et al., 2014b) as the benchmark corpora for experimentation. Here, we give a brief introduction to the two corpora:

- The RST-DT corpus contains 385 news articles (347 for training and 38 for testing) from the Wall Street Journal (WSJ). Following previous work, we randomly select 34 documents from the training corpus as the development corpus for parameter tuning. And we also binarize those non-binary subtrees in RST-DT with right-branching (Sagae and Lavie, 2005) for preprocessing.

- The Chinese CDTB corpus is motivated by taking advantages of both the English RST-DT corpus and the PDTB corpus (Prasad et al., 2008). The CDTB corpus annotates each paragraph as a Connective-driven Discourse Tree (CDT). The corpus consists of 500 newswire articles which are further segmented into 2336 paragraphs and 10650 EDUs. The corpus is divided into three parts with 425 articles (2002 CDT trees) for training, 25 articles (105 CDT trees) for validation, and 50 articles (229 CDT trees) for testing.

Metrics. Following previous studies, we measure the performance of bare tree structure (**S**), tree structure labeled with nuclearity (**N**), and tree structure labeled with rhetorical relation (**R**). Recently, the Full (**F**) indicator is used to estimate the tree structure labeled with both nuclearity and relation categories. However, since current performances on S, N and R are imbalanced, the performance on F is much limited by relation prediction. In other words, the Full score may underestimate the performance in span and nuclearity prediction. In this work, we combine nuclearity and rhetorical relation tags for joint N-R prediction aiming to reduce the uncertainty of the Full measure. Moreover, since RST-Parseval (Marcu, 2000) overestimates the DRS parsing performance to a certain extent, (Morey et al., 2017; Mabona et al., 2019; Zhang et al., 2020; Koto et al., 2021) adopt the original Parseval to reveal the actual performance level of DRS parsing. Following these studies, we also use the original Parseval for evaluation and report the micro-averaged F_1 scores by default.

Hyper-Parameter Setting. For word representation, we employed the 300D vectors of GloVe (Pennington et al., 2014) and the 1024D vectors of ELMo (Peters et al., 2018) for RST-DT and the 300D vectors of Qiu et al. (2018) (Qiu-W2V) for CDTB, and we did not update these vectors during training. The English POS tags were obtained through the Stanford CoreNLP toolkit (Manning et al., 2014), the Chinese tags were borrowed from Chinese PTB, and all the POS embeddings were optimized during training. For model learning, we used the development set to fine-tune the parameters in Table 1, and the number of parameter search trials was around 20. All the experiments based on the above-mentioned settings were conducted on GeForce RTX 2080Ti GPU, and the codes will be published at `https://github.com/NLP-Discourse-SoochowU/GAN_DP`.

Parameter	EN	CN
POS embedding	30	30
Uni-directional GRU	512	512
BiGRU	256	256
Biaffine-MLP-Split	128	64
Biaffine-MLP-NR	128	128
Boundary feature size	30	-
Dropout rate	0.2	0.33
Warm up epochs	7	7
Training epochs	20	20
Batch size (DTs)	5	64
Learning rate of D	1e-4	5e-4
Learning rate of other nets	1e-3	1e-3
α_1	0.3	0.3
α_2	1.0	1.0

Table 1: Fine-tuned hyper-parameters.

	Systems	S	N	R	F
EN	T2D	70.7	58.3	46.5	**45.2**
	+ DS	69.2	57.7	46.1	44.9
	+ TC	70.6	57.9	46.1	44.4
CN	T2D	82.5	57.3	51.7	48.2
	+ DS	83.2	57.8	52.7	**49.0**
	+ DS&TC	85.2	57.3	53.3	45.7

Table 2: Results under different model settings. "T2D" denotes our simplified architecture, which excludes the dummy split points and only uses two classifiers for DRS parsing; "DS" means the dummy split points are used; "TC" means three classifiers are used.

5.2 Experimental Results

Comparison between different system settings. As stated before, we explore to make possible improvements to the top-down architecture of Zhang et al. (2020). Here, we study the effects of these simplification methods based on our simplified architecture. For clarity, we remove the adversarial learning process in each system, and the results are presented in Table 2. For the RST-DT corpus, the first two rows show that the top-down parser performs worse when dummy split points are used, and the decline is obvious in tree structure parsing. Then, we further apply three classifiers to the simplified architecture, and the results (lines 1 and 3) show that the Full score drops by 1.8% for lack of correlation between the three learning goals. For the CDTB corpus, due to the differences in languages and annotation strategies, the situation is quite different. Specifically, lines 4 and 5 show that the top-down parser performs better on all the four indicators when using dummy split points (Zhang et al., 2020). Based on the better-performing parser using "DS", we further report its performance with three independent classifiers used, and the results (line 6) show that the Full score still drops a lot (6.7%), which suggests the necessity of joint N-R prediction. Considering the above results, in the following, we separately use two sets of model settings for different languages. For English, we build our final model based on the simplified architecture without dummy split points. For Chinese, we build our final model based on the architecture of Zhang et al. (2020). For both systems, we only use two classifiers for DRS parsing.

	Systems	S	N	R	F
EN	Final	71.8	59.5	47.0	45.9
	- *Advers. bot*	70.7	58.3	46.5	45.2
CN	Final	84.9	58.4	54.5	50.3
	- *Advers. bot*	83.2	57.8	52.7	49.0

Table 3: Comparison on the adversarial bot.

	Systems	S	N	R	F
EN	JE2017-updated	64.1	54.2	46.8	**46.3**
	Feng and Hirst (2014)	68.6	55.9	45.8	44.6
	Li et al. (2016)	64.5	54.0	38.1	36.6
	Braud et al. (2016)	59.5	47.2	34.7	34.3
	Braud et al. (2017)	62.7	54.5	45.5	45.1
	Mabona et al. (2019)	67.1	57.4	45.5	45.0
	Zhang et al. (2020)	67.2	55.5	45.3	44.3
	Ours (GloVe)	69.9	57.3	46.3	45.0
	Ours (ELMo)	**71.8**	**59.5**	**47.0**	45.9
CN	*Zhang et al. (2020)*	*85.2*	*57.3*	*53.3*	*45.7*
	Zhang et al. (2020)*	84.0	**59.0**	54.2	47.8
	Ours (Qiu-W2V)	**84.9**	58.4	**54.5**	**50.3**

Table 4: Performance comparison with previous work. Results of the first five lines are directly borrowed from (Morey et al., 2017). "*" denotes the updated results based on the strict evaluation metric we use.

Comparison on the adversarial bot. Here, we perform experiments to explore the effects of the adversarial learning approach, and the experimental results are presented in Table 3. For the RST-DT corpus, the results show that our adversarial model setting can improve the performance on all the four indicators, especially in structure and nuclearity prediction. Similarly, the results on the CDTB corpus show that our adversarial method still works much better than the unreinforced parser in structure, relation, and full detection. The overall results indicate that the global optimization method we use is definitely effective, although the effectiveness has not yet reached the level of qualitative change. In fact, as a preliminary attempt for global optimization of DRS parsing, this research still has much room for improvement which deserves further exploration.

Comparison with previous studies. In this part, we compare with seven previous state-of-the-art (SOTA) parsers on text-level DRS parsing. Here, we briefly review these studies as follows:

- Ji and Eisenstein (2014), a shift-reduce parser with an SVM that is trained by their extracted latent features. In this paper, we compare with the updated version of their parser (designated as "JE2017-updated") (Morey et al., 2017).

- Feng and Hirst (2014), a two-stage greedy parser with linear-chain CRF models and some hand-engineered features.

- Li et al. (2016), an attention-based hierarchical neural model with hand-crafted features used.

- Braud et al. (2016), a hierarchical BiLSTM model that leverages information from various sequence prediction tasks.

- Braud et al. (2017), a transition-based neural model with both cross-lingual information and hand-crafted features used.

- Mabona et al. (2019), a generative model with a beam search algorithm used for DRS parsing.

- Zhang et al. (2020), a top-down neural architecture tailored for text-level DRS parsing. Different from many previous studies, this parser is a pure neural parser without using any additional hand-crafted features.

For the RST-DT corpus, the results are presented in the upper part of Table 4. From the results, although our previous top-down parser (Zhang et al., 2020) can achieve good results without using hand-crafted features, the performance is still far from perfect. Comparing our GloVe-based top-down parser with previous state-of-the-art parsers, our parser performs better than most previous ones due to its ability in leveraging global context and the adversarial learning strategy. Furthermore, comparing the final parser (line 9) with previous work, our ELMo-based parser can further improve the performance on all the four indicators, and the improvements on structure (4.7%) and nuclearity (3.7%) are significant. Obviously, the contextualized word representation can greatly improve the parsing performance, especially in such a task with small-scale data corpora.

For the CDTB corpus, we explore to employ a more strict metric[4] for performance evaluation and the overall results are presented in the lower part of Table 4. In comparison with previous work, our parser achieves comparable performance in nuclearity and relation prediction and much better results on the other two indicators, which proves the usefulness of the adversarial nets we use. In

[4] We borrow the `strict` evaluation method from `https://github.com/NLP-Discourse-SoochowU/t2d_discourseparser` for evaluation in this study, and report the macro-averaged F1-scores for performance.

	Systems	S	N	R	F
EN	Koto et al. (2021)	73.1	62.3	51.5	50.3
	Ours (XLNet)	**76.3**	**65.5**	**55.6**	**53.8**
	- *Advers. bot*	76.1	64.4	54.3	52.9
CN	Ours (Qiu-W2V)	84.9	58.4	54.5	50.3
	Ours (XLNet)	**86.6**	**65.0**	**62.1**	**55.4**
	- *Advers. bot*	85.8	64.5	60.5	53.7

Table 5: Performance comparison with LMs used.

Systems	UAS	LAS
Wang et al. (2017)*	61.5	47.8
Yu et al. (2018)*	61.9	48.4
Kobayashi et al. (2020)*	64.9	48.5
Ours (Final)	**72.3**	**57.6**
- *Advers. bot*	71.4	56.5

Table 6: Evaluation on dependency trees. "*" denotes the results are borrowed from (Kobayashi et al., 2020).

particular, compared with previous parsers, our parser performs significantly better on "F" due to the joint prediction of nuclearity and relation categories. This suggests the robustness of our simplified parser with only two classifiers. Moreover, since the two top-down DRS parsers in the table show similar results on "R", we speculate that the Chinese rhetorical relation prediction has encountered a bottleneck to some extent, which requires more effort to be invested.

Performances based on the SOTA language models. Recently, more and more researchers (Shi et al., 2020; Koto et al., 2021) propose to improve DRS parsing performance through powerful language models (LMs) like Bert (Devlin et al., 2019) and XLNet (Yang et al., 2019). Following these studies, in this work, we perform additional experiments on the XLNet-base models in (Yang et al., 2019) and (Cui et al., 2020) for the RST-DT and CDTB corpus, respectively. For better model integration, we slightly adjust the previously described model architecture[5], more specifically, the EDU encoder. We first use a pre-trained LM to encode each entire discourse where each EDU is attached with the `[SEP]` and `[CLS]` tokens and then take the LM outputs corresponding to `[CLS]` as our EDU representation. Moreover, we segment each document according to the maximum length of 768 tokens and encode these text segments one by one to avoid the problem of memory overflow.

For the RST-DT corpus, we report the results of the recent Bert-based top-down parser (Koto et al., 2021) for comparison. For the CDTB corpus, we compare with our previously described system based on traditional word vectors, and the overall results are shown in Table 5. From the results we find that our parsers achieve superior results when using the contextualized XLNet for experimentation, which suggests the great effectiveness of pre-trained LMs in such a task with limited corpus size. Moreover, the ablation study on the adversarial learning strategy further demonstrates the usefulness of our proposed method. It should be noted that we report the performance using LMs in this paper never mean to advocate using pre-trained LMs or blindly pursuing performance improvements in DRS parsing. Sometimes, the rewards generated by the large-scale LMs could be quite different from and much more effective than that generated by language phenomena, which may hinder the study on the relatively shallow (compared with powerful LMs) yet valuable discourse features. With this in mind, it is reasonable to perform ablation study using simple word representation to explore useful discourse features and report the performance on powerful LMs for reference.

5.3 Analysis and Discussion

Performance Evaluation of Dependency Trees. Recently, discourse-level dependency structure has attracted more and more attention. Here, we explore whether the proposed global optimization method can improve the RST dependency analysis to some extent. To achieve this, we first convert the predicted DRS trees into dependency trees as Kobayashi et al. (2020) did and then perform evaluation on the converted dependencies labeled (LAS) and unlabeled (UAS) with rhetorical relations, and the results are shown in Table 6. Firstly, lines 1 to 4 show that our parser can greatly outperform previous systems in terms of both UAS and LAS indicators. Secondly, the last two rows show that the global optimization of constituency trees can simultaneously improve the dependency performance, which further proves the usefulness of our proposed adversarial method.

Remarkable Progress in DRS Parsing. Compared with Chinese DRS parsing where each paragraph is annotated as a DT, the English parsing with 313 DTs for training is much more challenging. Nevertheless, results in Table 4 and Table 5 show that our parser can largely outperform previous

[5] Adjusted model parameters are shown in Appendix.

Systems	NN/23%	NS/61%	SN/16%
Ours (GloVe)	43.3	62.9	55.7
Ours (ELMo)	47.8	64.1	58.5
Ours (XLNet)	56.7	**67.4**	**69.6**
- *Advers. bot*	**58.8**	66.4	66.7

Table 7: Performance on nuclearity detection.

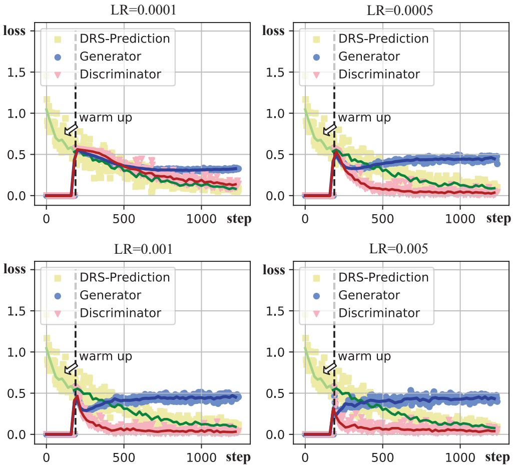


Figure 5: Convergence of our parsing model over different learning rates (LRs).

state-of-the-art parsers on “Full”. (i) For nuclearity prediction, we display the results of our parsers on each nuclearity category to explore where the improvement comes from, as shown in Table 7. From the results, it’s obvious that the LM we use plays a big role in nuclearity prediction, and the proposed adversarial method can further improve the performance to a certain extent. (ii) For relation prediction, the classification problem with 18 coarse-grained relation tags (RST-DT) is really a challenge. From the results in Table 4 we can find that the progress in relation prediction is much limited in recent decade for the lack of data. And most of previous state-of-the-art parsers employee a variety of hand-engineered features for good performance. Hopefully, the experimental results in Table 5 show that powerful LMs can free data-driven models from corpus size limitation and thus our XLNet-based parser strongly outperforms JE2017-updated (Morey et al., 2017) by 18.8% on “R”. The results of our parsers on each rhetorical relation category are shown in Appendix.

Discussion on Adversarial Learning. Similar to previous GAN work, improving the quality of the generated tree images is really a challenge, and the instability of the adversarial learning process is another intractable issue. In order for our model to continuously modify the generated images even when they are correctly classified, we leverage a least squares loss in our system for model learning. To avoid the over-learning of the discriminator, we tune it with a moderate learning rate and parameter scale. Intuitively, the convergence of our model over different learning rates is presented in Figure 5. From the results, as the learning rate of the discriminator increases, the fluctuation of the loss value becomes larger, and it is hard to reduce the generator loss. In these four cases, the first group seems to be more stable and in line with our expectations. Therefore, we set the learning rate to 1e-4 in our systems for experimentation. Notably, we also tried the sigmoid cross entropy loss in this research which performs much worse than the LSGAN we use. For reference, we also present the model convergence over different loss functions in Appendix for reference.

6 Conclusion

In this research, we explored a global optimization method based on recent top-down frameworks. Particularly, we proposed a novel strategy to transform both gold standard and predicted DRS trees into tree diagrams with two color channels. On this basis, we produced an LSGAN-based adversarial bot between gold and fake trees for global optimization. Experimental results on two popular corpora showed that our proposed adversarial approach is effective in DRS parsing and has established new state-of-the-art results for both corpora.

Acknowledgements

Here, the first author (Longyin Zhang) would like to thank his fiancee, Dr. Xin Tan, for her valuable discussion on this research. This work was supported by the National Key R&D Program of China under Grant No. 2020AAA0108600, Projects 61876118 and 61976146 under the National Natural Science Foundation of China and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

Chloé Braud, Maximin Coavoux, and Anders Søgaard. 2017. Cross-lingual RST discourse parsing. In *EACL*, pages 292–304, Valencia, Spain. Association for Computational Linguistics.

Chloé Braud, Barbara Plank, and Anders Søgaard. 2016. Multi-view and multi-task training of RST discourse parsers. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1903–1913, Osaka, Japan. The COLING 2016 Organizing Committee.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Francine Chen and Yan-Ying Chen. 2019. Adversarial domain adaptation using artificial titles for abstractive title generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2197–2203, Florence, Italy. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *The 5th International Conference on Learning Representations, ICLR2017*.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, Brussels, Belgium. Association for Computational Linguistics.

Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521, Baltimore, Maryland. Association for Computational Linguistics.

Hugo Hernault, Helmut Prendinger, Mitsuru Ishizuka, et al. 2010. Hilda: A discourse parser using support vector machine classification. *Dialogue and Discourse*, 1(3).

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, Baltimore, Maryland. Association for Computational Linguistics.

Yangfeng Ji and Noah A. Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005, Vancouver, Canada. Association for Computational Linguistics.

Shafiq Joty, Giuseppe Carenini, Raymond Ng, and Yashar Mehdad. 2013. Combining intra- and multi-sentential rhetorical parsing for document-level discourse analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 486–496, Sofia, Bulgaria. Association for Computational Linguistics.

Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2020. Top-down rst parsing utilizing granularity levels in documents. In *Association for the Advancement of Artificial Intelligence 2020, AAAI2020*.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2021. Top-down discourse parsing via sequence labelling. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 715–726, Online. Association for Computational Linguistics.

Jiwei Li, Rumeng Li, and Eduard Hovy. 2014a. Recursive deep models for discourse parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2061–2069, Doha, Qatar. Association for Computational Linguistics.

Qi Li, Tianshi Li, and Baobao Chang. 2016. Discourse parsing with attention-based hierarchical neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 362–371, Austin, Texas. Association for Computational Linguistics.

Yancui Li, wenhe Feng, jing Sun, Fang Kong, and Guodong Zhou. 2014b. Building chinese discourse corpus with connective-driven dependency tree structure. In *Proceedings of EMNLP 2014*, pages 2105–2114.

Xiang Lin, Shafiq Joty, Prathyusha Jwalapuram, and M Saiful Bari. 2019. A unified linear-time framework for sentence-level discourse parsing. In *ACL*, pages 4190–4200, Florence, Italy. Association for Computational Linguistics.

Linlin Liu, Xiang Lin, Shafiq Joty, Simeng Han, and Lidong Bing. 2019. Hierarchical pointer net parsing. In *EMNLP-IJCNLP*, pages 1006–1016, Hong Kong, China. Association for Computational Linguistics.

Amandla Mabona, Laura Rimell, Stephen Clark, and Andreas Vlachos. 2019. Neural generative rhetorical structure parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference* *on Natural Language Processing (EMNLP-IJCNLP)*, pages 2284–2295, Hong Kong, China. Association for Computational Linguistics.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. 2017. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821.

Daniel Marcu. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA.

Mathieu Morey, Philippe Muller, and Nicholas Asher. 2017. How much progress have we made on RST discourse parsing? a replication study of recent results on the RST-DT. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1319–1324, Copenhagen, Denmark. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The penn discourse treebank 2.0. In *LREC 2008*.

Yuanyuan Qiu, Hongzheng Li, Shen Li, Yingdi Jiang, Renfen Hu, and Lijiao Yang. 2018. Revisiting correlations between intrinsic and extrinsic evaluations of word embeddings. In *CCL & NLP-NABD 2017*, pages 209–221. Springer.

Kenji Sagae and Alon Lavie. 2005. A classifier-based parser with linear run-time complexity. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 125–132, Vancouver, British Columbia. Association for Computational Linguistics.

Ke Shi, Zhengyuan Liu, and Nancy F. Chen. 2020. An end-to-end document-level neural discourse parser exploiting multi-granularity representations. *CoRR*, abs/2012.11169.

Yizhong Wang, Sujian Li, and Houfeng Wang. 2017. A two-stage parsing method for text-level discourse analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 184–188, Vancouver, Canada. Association for Computational Linguistics.

Jiawei Wu, Xin Wang, and William Yang Wang. 2019. Self-supervised dialogue learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3857–3867, Florence, Italy. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Nan Yu, Meishan Zhang, and Guohong Fu. 2018. Transition-based neural RST parsing with implicit syntax features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 559–570, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Longyin Zhang, Yuqing Xing, Fang Kong, Peifeng Li, and Guodong Zhou. 2020. A top-down neural architecture towards text-level parsing of discourse rhetorical structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6386–6395, Online. Association for Computational Linguistics.

Wei Zou, Shujian Huang, Jun Xie, Xinyu Dai, and Jiajun Chen. 2020. A reinforced generation of adversarial examples for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3486–3497, Online. Association for Computational Linguistics.

Appendix

A. Adversarial Model Learning

Here, we display the convergence of our models with different loss functions and model settings applied, as shown in Figure 6. Comparing the first two legends, since the sigmoid cross entropy loss suffers from gradient vanishing, it's hard for our model to update the generator net, and the generator loss keeps growing up. To avoid the over-learning of the discriminator net, we simplify the original discriminator network from a 3-layer MLP to a linear function, and the results are presented in Figure 6 (c). From the results, it's really hard to train both generator and discriminator nets, and the adversarial learning in Figure 6 (c) seems to be meaningless for DRS parsing.

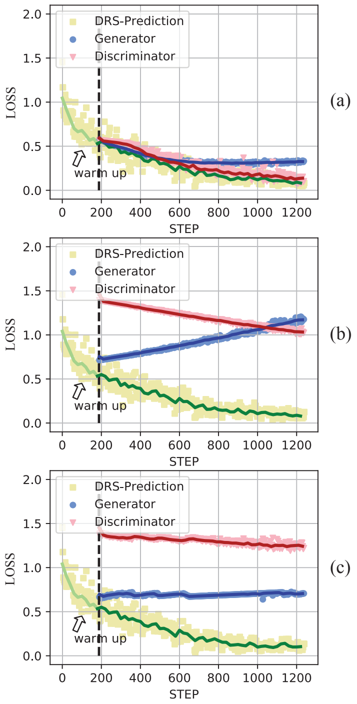


Figure 6: Figure (a) refers to our final model based on LSGAN; figure (b) refers to our model with the sigmoid cross entropy loss function used; based on figure (b), we use a simplified discriminator in figure (c).

B. Results on Different Relation Categories

Table 8 and Table 9 present the performances (F1-scores) of our systems on each relation category in the RST-DT and CDTB corpora, respectively.

C. Configurations of the LM-based Systems

For better model integration, we slightly tuned the model hyper-parameters to adapt to the LM-based systems. For RST-DT, we set the LRs of all the nets to 1e-4, the hidden size of BiGRU to 384, the hidden size of uni-directional GRU to 768, and the batch size to 1 to suit the NVIDIA Tesla P40 GPU memory. For CDTB, we set the LRs of the discriminator, LM, and other nets to 5e-4, 1e-4, and 2e-5, respectively. We trained the LM-based systems for around 30 rounds and the other system settings remained the same as the aforementioned non-LM-based systems.

Type-ratio%	GloVe	ELMo	XLNet
Elaborate-30.4	47.9	48.8	60.4
Joint-15.1	36.3	39.2	49.4
Attribution-11.7	77.9	83.0	86.7
Same-unit-10.9	70.3	71.9	75.9
Contrast-5.8	34.5	27.0	42.6
Explanation-3.8	11.3	16.1	21.7
Background-3.4	23.0	20.8	27.8
Temporal-3.0	15.4	15.5	34.6
Cause-2.9	3.7	7.7	18.5
Evaluation-2.2	4.1	0.0	10.5
Enablement-2.2	54.7	42.0	66.7
Comparison-1.7	12.5	12.9	36.7
Topic-change-1.6	7.7	11.1	40.0
Textual-org-1.3	20.0	28.6	53.3
Condition-1.2	42.1	29.0	62.5
Topic-comment-1.0	0.0	0.0	8.3
Manner-means-0.8	33.3	32.1	44.0
Summary-0.8	47.8	44.0	50.0

Table 8: Results on the RST-DT corpus. "ratio" means the proportion of each category label in the corpus.

Type-ratio%	Qiu-W2V	XLNet
并列 / Same-unit-47.8	80.2	88.0
解说 / Explanation-12.6	50.0	60.7
因果 / Cause-9.4	32.5	55.9
顺承 / Consequent-7.1	4.1	58.1
目的 / Purpose-4.6	48.5	58.5
例证 / Example-3.4	10.5	34.5
总分 / Overall-branch-3.2	75.0	73.9
评价 / Evaluation-3.1	26.7	56.3
转折 / Contrast-2.7	69.0	75.0
背景 / Background-1.8	0.0	36.4
条件 / Condition-1.0	0.0	16.7
假设 / Suppose-1.0	0.0	66.7
递进 / Progressive-0.9	0.0	0.0
对比 / Comparison-0.8	0.0	40.0
推断 / Deduce-0.5	0.0	0.0
让步 / Concession-0.2	0.0	0.0

Table 9: Results on the CDTB corpus.